What is claimed is:

1. A method of effecting secure communications between a server and a client, the method comprising:

detecting a client connection at a first port;

providing the client with a decoy port number; and

providing services to the client on a second port that is mapped to the decoy port number.

2. A method as defined in Claim 1, wherein the decoy port number is provided to the client by the operation of a routine that is associated with the server.

3. A method as defined in Claim 2, further comprising:

launching the server on the second port; and

monitoring the second port for a connection by the client.

4. A method as defined in Claim 3, further comprising;

if there is no connection by the client within a predetermined time interval, terminating execution of the server on the second port.

5. A method as defined in Claim 2, further comprising:

maintaining a table of available decoy port numbers that are mapped to valid port numbers.

6. A method as defined in Claim 5, further comprising:

subsequent to providing the decoy port number to the client, launching the server on the second port.

7. A method as defined in Claim 6, further comprising:

monitoring the second port for a connection by the client, and

if there is no connection by the client within a predetermined time interval, terminating execution of the server on the second port.

8. A method as defined in Claim 7, further comprising:

verifying via a server log file whether the client effected a connection to the second port within the predetermined time interval.

9. A computer system comprising:

a plurality of ports, each port having a respective port number;

a server application; and

a routine that, if executed, is operative to:

detect a client connection at a first port;

provide the client with a decoy port number; and

provide services to the client on a second port that is mapped to the decoy port number.

10. A computer system as defined in Claim 9, wherein the routine, if executed, is operative to:

launch the server application on the second port; and

monitor the second port for a connection by the client.

11. A computer system as defined in Claim 10, wherein the routine, if executed, is operative to terminate execution of the server application on the second port if there is no connection by the client within a predetermined time interval.

12. A computer system as defined in Claim 9, wherein the routine, if executed, is operative to maintain a table of decoy port numbers and wherein each of a plurality of decoy port numbers and is mapped to a valid port number.

13. A computer system as defined in Claim 12, wherein the routine, if executed, is operative to:

launch the server application on the second port subsequent to providing the decoy port number to the client.

14. A computer system as defined in Claim 13, wherein the routine, if executed, is operative to:

monitor the second port for a connection by the client; and

if there is no connection by the client within a predetermined time interval, terminate execution of the server on the second port.

15. A server computer system comprising:

a plurality of ports, each port having a respective port number;

a first server application; and

a first routine that is associated with the first server application and that, if executed, is operative to:

detect a client connection at a first port;

provide the client with a decoy port number;

terminate the connection to the first port; and

provide services to the client on a second port that is mapped to the decoy port number;

a second server application; and

a second routine that is associated with the second server application and that, if executed, is operative to:

detect a client connection at a third port;

provide the client with a decoy port number;

terminate the connection to the third port; and

provide services to the client on a fourth port that is mapped to the decoy port number.

16. A server computer as defined in Claim 15, wherein the first routine and the second routine, if executed are operable, respectively, to:

terminate execution of the first server application on the second port if there is no client connection within a predetermined time interval; and

terminate execution of the second server application on the fourth port if there is no client connection within a predetermined time interval.

17. A method comprising:

attempting to access a server application on a first port;

receiving a decoy port number;

translating the decoy port number to a translated port number; and

connecting to the server application on the translated port number.

18. A method as defined in Claim 17, wherein the decoy port number is translated using a wrapper script associated with a client application.

19. A method as defined in Claim 17, wherein the decoy port number is translated using code embedded in a client application.

20. A method as defined in Claim 17, further comprising:

mapping the decoy port number to an intermediate port number; and

effecting an offset to the intermediate port number.

21. A computer system comprising:

a plurality of ports, each port having a respective port number;

an application; and

means for effecting secure access to the application by redirecting a client from a first port to a second port.

22. A computer system as defined in Claim 21, wherein the means for effecting secure access comprises:

a routine that, if executed, is operable to provide the client with a decoy port number that maps to the second port number.

23. An article comprising a machine-readable storage medium that comprises instructions that, if executed, are operable to:

detect a connection at a first port by a client application;

provide the client application with a decoy port number; and

cause a server application to be launched at a second port that is mapped to the decoy port number.

24. An article as defined in Claim 23, further comprising instructions, that, if executed, are operable to:

monitor the second port; and

if there is no connection by the client application within a predetermined time interval, terminate execution of the server on the second port.

25. An article as defined in Claim 23, wherein the storage medium further comprises a table of decoy port numbers that are mapped to valid port numbers.

26. An article as defined in Claim 25, further comprising instructions, that, if executed, are operable to:

monitor the second port; and

if there is no connection by the client application within a predetermined time interval, terminate execution of the server application on the second port.

27. A client/server architecture comprising:

a server computer system; and

a server application installed on the sever computer system and comprising instructions that, if executed on the server computer system, are effective to:

detect a connection at a first port by a client application;

provide the client application with a decoy port number;

terminate the connection on the first port; and

provide services to the client application on a second port that is mapped to the decoy port number.

28. A client/server architecture as defined in Claim 27, further comprising:

a client computer system; and

a client application installed on the client computer system and comprising instructions that, if executed on the client computer system, are effective to:

attempt to access the server application on the first port;

translate the decoy port number to the second port number; and

connect to the server application on the second port.

29. A client/server architecture as defined in Claim 28, wherein the server application comprises instructions that, if executed by the server computer system are effective to:

launch the server application on the second port;

monitor the second port for a connection by the client; and

terminate execution of the server application on the second port if there is no connection by the client application within a predetermined time interval.

30. A client/server architecture as defined in Claim 28, wherein the client application further comprises instructions that, if executed on the client computer system, are effective to:

map the decoy port number to an intermediate port number; and

impart an offset to the intermediate port number so as to derive the second port number.